

學號：L091197 姓名：郭紘安

SDN-NFV Lab 2

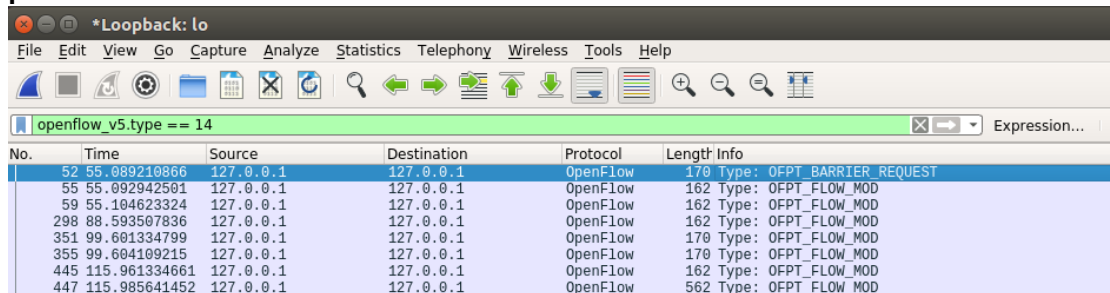
Part 1: Answers Questions

1. How many OpenFlow headers of type “OFPT_FLOW_MOD” are there among all the packets?

Ans:

No.	52	55	59	298	351	355	445	447
OFPT_FLOW_MOD	1	1	1	1	1	1	1	5

Total : 12 OpenFlow headers of type “OFPT_FLOW_MOD” among all packets



No.	Time	Source	Destination	Protocol	Length	Info
52	55.089210866	127.0.0.1	127.0.0.1	OpenFlow	170	Type: OFPT_BARRIER_REQUEST
55	55.092942501	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
59	55.104623324	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
298	88.593507836	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
351	99.601334799	127.0.0.1	127.0.0.1	OpenFlow	170	Type: OFPT_FLOW_MOD
355	99.604109215	127.0.0.1	127.0.0.1	OpenFlow	170	Type: OFPT_FLOW_MOD
445	115.961334661	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
447	115.985641452	127.0.0.1	127.0.0.1	OpenFlow	562	Type: OFPT_FLOW_MOD

```
▶ Frame 52: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface 0
▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 6653, Dst Port: 58632, Seq: 1601, Ack: 1909, Len: 104
▼ OpenFlow 1.4
  Version: 1.4 (0x05)
  Type: OFPT_BARRIER_REQUEST (20)
  Length: 8
  Transaction ID: 18
```

2. What are the matching fields and the corresponding actions in each of “OFPT_FLOW_MOD”

Ans:

● No.52

Match

Type: OFPMT_OXM (1)

Length: 10

OXM field

Class: OFPXM_OPENFLOW_BASIC (0x8000)

0000 101. = Field: OFPXM_OFB_ETH_TYPE (5)

.... ...0 = Has mask: False

Length: 2

Value: 802.1 Link Layer Discovery Protocol (LLDP) (0x88cc)

Pad: 000000000000

Action

Type: OFPAT_OUTPUT (0)

Length: 16

Port: OFPP_CONTROLLER (4294967293)

Max length: OFPCML_NO_BUFFER (65535)

Pad: 000000000000

● **No.55**

Match

Type: OFPMT_OXM (1)
Length: 10
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 101. = Field: OFPXMT_OFB_ETH_TYPE (5)
.... ...0 = Has mask: False
Length: 2
Value: Unknown (0x8942)
Pad: 000000000000

Action

Type: OFPAT_OUTPUT (0)
Length: 16
Port: OFPP_CONTROLLER (4294967293)
Max length: OFPCML_NO_BUFFER (65535)
Pad: 000000000000

● **No.59**

Match

Type: OFPMT_OXM (1)
Length: 10
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 101. = Field: OFPXMT_OFB_ETH_TYPE (5)
.... ...0 = Has mask: False
Length: 2
Value: ARP (0x0806)
Pad: 000000000000

Action

Type: OFPAT_OUTPUT (0)
Length: 16
Port: OFPP_CONTROLLER (4294967293)
Max length: OFPCML_NO_BUFFER (65535)
Pad: 000000000000

● **No.298**

Match

Type: OFPMT_OXM (1)
Length: 10
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 101. = Field: OFPXMT_OFB_ETH_TYPE (5)
.... ...0 = Has mask: False
Length: 2
Value: IPv4 (0x0800)
Pad: 000000000000

Action

Type: OFPAT_OUTPUT (0)
Length: 16
Port: OFPP_CONTROLLER (4294967293)
Max length: OFPCML_NO_BUFFER (65535)
Pad: 000000000000

● **No.351**

Match

Type: OFPMT_OXM (1)

Length: 32

OXM field

Class: OFPXM_OPENFLOW_BASIC (0x8000)

0000 000. = Field: OFPXMT_OFB_IN_PORT (0)

.... ...0 = Has mask: False

Length: 4

Value: 2

OXM field

Class: OFPXM_OPENFLOW_BASIC (0x8000)

0000 011. = Field: OFPXMT_OFB_ETH_DST (3)

.... ...0 = Has mask: False

Length: 6

Value: 42:d1:f2:db:21:45 (42:d1:f2:db:21:45)

OXM field

Class: OFPXM_OPENFLOW_BASIC (0x8000)

0000 100. = Field: OFPXMT_OFB_ETH_SRC (4)

.... ...0 = Has mask: False

Length: 6

Value: ce:07:03:b3:74:44 (ce:07:03:b3:74:44)

Action

Type: OFPAT_OUTPUT (0)

Length: 16

Port: 1

Max length: 0

Pad: 000000000000

● **No.355**

Match

Type: OFPMT_OXM (1)

Length: 32

OXM field

Class: OFPXM_OPENFLOW_BASIC (0x8000)

0000 000. = Field: OFPXMT_OFB_IN_PORT (0)

.... ...0 = Has mask: False

Length: 4

Value: 1

OXM field

Class: OFPXM_OPENFLOW_BASIC (0x8000)

0000 011. = Field: OFPXMT_OFB_ETH_DST (3)

.... ...0 = Has mask: False

Length: 6

Value: ce:07:03:b3:74:44 (ce:07:03:b3:74:44)

OXM field

Class: OFPXM_OPENFLOW_BASIC (0x8000)

0000 100. = Field: OFPXMT_OFB_ETH_SRC (4)

.... ...0 = Has mask: False

Length: 6

Value: 42:d1:f2:db:21:45 (42:d1:f2:db:21:45)

Action

Type: OFPAT_OUTPUT (0)

Length: 16

Port: 2

Max length: 0

Pad: 000000000000

● **No.445**

Match

Type: OFPMT_OXM (1)
Length: 10
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 101. = Field: OFPXMT_OFB_ETH_TYPE (5)
.... ...0 = Has mask: False
Length: 2
Value: 802.1 Link Layer Discovery Protocol (LLDP) (0x88cc)
Pad: 000000000000

Action

Type: OFPAT_OUTPUT (0)
Length: 16
Port: OFPP_CONTROLLER (4294967293)
Max length: OFPCML_NO_BUFFER (65535)
Pad: 000000000000

● **No.447**

Match

Type: OFPMT_OXM (1)
Length: 10
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 101. = Field: OFPXMT_OFB_ETH_TYPE (5)
.... ...0 = Has mask: False
Length: 2
Value: Unknown (0x8942)
Pad: 000000000000

Action

Type: OFPAT_OUTPUT (0)
Length: 16
Port: OFPP_CONTROLLER (4294967293)
Max length: OFPCML_NO_BUFFER (65535)
Pad: 000000000000

Match

Type: OFPMT_OXM (1)
Length: 32
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 000. = Field: OFPXMT_OFB_IN_PORT (0)
.... ...0 = Has mask: False
Length: 4
Value: 1
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 011. = Field: OFPXMT_OFB_ETH_DST (3)
.... ...0 = Has mask: False
Length: 6
Value: ce:07:03:b3:74:44 (ce:07:03:b3:74:44)
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 100. = Field: OFPXMT_OFB_ETH_SRC (4)
.... ...0 = Has mask: False
Length: 6
Value: 42:d1:f2:db:21:45 (42:d1:f2:db:21:45)

Action

Type: OFPAT_OUTPUT (0)
Length: 16
Port: 2
Max length: 0
Pad: 000000000000

Match

Type: OFPMT_OXM (1)
Length: 10
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 101. = Field: OFPXMT_OFB_ETH_TYPE (5)
.... ...0 = Has mask: False
Length: 2
Value: ARP (0x0806)
Pad: 000000000000

Action

Type: OFPAT_OUTPUT (0)
Length: 16
Port: OFPP_CONTROLLER (4294967293)
Max length: OFPCML_NO_BUFFER (65535)
Pad: 000000000000

Match

Type: OFPMT_OXM (1)
Length: 32
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 000. = Field: OFPXMT_OFB_IN_PORT (0)
.... ...0 = Has mask: False
Length: 4
Value: 2
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 011. = Field: OFPXMT_OFB_ETH_DST (3)
.... ...0 = Has mask: False
Length: 6
Value: 42:d1:f2:db:21:45 (42:d1:f2:db:21:45)
OXM field
Class: OFPXM_OPENFLOW_BASIC (0x8000)
0000 100. = Field: OFPXMT_OFB_ETH_SRC (4)
.... ...0 = Has mask: False
Length: 6
Value: ce:07:03:b3:74:44 (ce:07:03:b3:74:44)

Action

Type: OFPAT_OUTPUT (0)
Length: 16
Port: 1
Max length: 0
Pad: 000000000000

```

Match
  Type: OFPMT_OXM (1)
  Length: 10
  OXM field
    Class: OFPXM_OPENFLOW_BASIC (0x8000)
    0000 101. = Field: OFPXMT_OFB_ETH_TYPE (5)
    .... ...0 = Has mask: False
    Length: 2
    Value: IPv4 (0x0800)
  Pad: 000000000000
Action
  Type: OFPAT_OUTPUT (0)
  Length: 16
  Port: OFPP_CONTROLLER (4294967293)
  Max length: OFPCML_NO_BUFFER (65535)
  Pad: 000000000000

```

3.What are the values of the priority fields of all “OFPT_FLOW_MOD” messages?

Ans:

- No.52
 - i.Priority: 40000
- No.55
 - i.Priority: 40000
- No.59
 - i.Priority: 40000
- No.298
 - i.Priority: 5
- No.351
 - i.Priority: 10
- No.355
 - i.Priority: 10
- No.445
 - i.Priority: 40000
- No.447
 - i.Priority: 40000
 - ii.Priority: 10
 - iii.Priority: 40000
 - iv.Priority: 10
 - v.Priority: 5

Part 2: Install Flow Rules

Step 1 : Deactivate all the apps, except those initially activated.

```
andy@root > apps -a -s
* 15 org.onosproject.optical-model      2.2.0    Optical Network Model
* 16 org.onosproject.drivers            2.2.0    Default Drivers
* 27 org.onosproject.openflow-base     2.2.0    OpenFlow Base Provider
* 28 org.onosproject.lldpprovider       2.2.0    LLDP Link Provider
* 29 org.onosproject.hostprovider       2.2.0    Host Location Provider
* 36 org.onosproject.openflow          2.2.0    OpenFlow Provider Suite
* 101 org.onosproject.gui2             2.2.0    ONOS GUI2
andy@root >
```

Step 2 : Start Mininet with default (minimal) topology
sudo mn --controller=remote,127.0.0.1:6653

```
andy@ubuntu:~/onos$ sudo mn --controller=remote,127.0.0.1:6653
[sudo] password for andy:
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Step 3 : Hand in flow rule files (.json)

```
curl -u onos:rocks -X POST -H 'Content-Type: application/json' -d @flows_s1-1_L091197.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
```

```
curl -u onos:rocks -X POST -H 'Content-Type: application/json' -d @flows_s1-2_L091197.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
```

```
andy@ubuntu:~/onos$ curl -u onos:rocks -X POST -H 'Content-Type: application/json' -d @flows_s1-1_L091197.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
```

```
andy@ubuntu:~/onos$ curl -u onos:rocks -X POST -H 'Content-Type: application/json' -d @flows_s1-2_L091197.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
```

Step 4 : h1 arping h2

h1 arping h2

```
mininet> h1 arping h2
ARPING 10.0.0.2 from 10.0.0.1 h1-eth0
Unicast reply from 10.0.0.2 [DA:16:CA:CB:E5:FF] 0.667ms
Unicast reply from 10.0.0.2 [DA:16:CA:CB:E5:FF] 0.564ms
^CSent 2 probes (1 broadcast(s))
Received 2 response(s)
```

Step 5 : Hand in flow rule files (.json)

```
curl -u onos:rocks -X POST -H 'Content-Type: application/json' -d @flows_s1-3_L091197.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
```

```
curl -u onos:rocks -X POST -H 'Content-Type: application/json' -d @flows_s1-4_L091197.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
```

```
andy@ubuntu:~/onos$ curl -u onos:rocks -X POST -H 'Content-Type: application/json' -d @flows_s1-3_L091197.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
```

```
andy@ubuntu:~/onos$ curl -u onos:rocks -X POST -H 'Content-Type: application/json' -d @flows_s1-4_L091197.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
```

Step 6 : h1 ping h2

h1 ping h2

```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.297 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.031 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3051ms
rtt min/avg/max/mdev = 0.031/0.105/0.297/0.111 ms
mininet>
```

[illegible]

Step 5 : CPUs utilization of VM

● Without using h1 ping h2

```
top - 09:27:48 up 6:41, 1 user, load average: 2.05, 1.99, 1.29
Tasks: 255 total, 3 running, 185 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.8 us, 0.8 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 98.3 si, 0.0 st
Mem: 8144668 total, 288348 free, 3200372 used, 4655948 buff/cache
Swap: 998396 total, 998128 free, 268 used. 4502672 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7	root	20	0	0	0	0	R	97.3	0.0	50:59.58	ksoftirqd+
16	root	20	0	0	0	0	R	95.7	0.0	51:12.27	ksoftirqd+
940	root	20	0	507304	111172	56128	S	1.7	1.4	6:12.94	Xorg
25451	andy	20	0	5228736	735720	25724	S	1.7	9.0	1:32.79	java
2191	andy	20	0	1528404	117912	70380	S	1.0	1.4	5:50.69	compiz
2403	andy	20	0	675632	44964	28372	S	1.0	0.6	1:15.19	gnome-ter+
3646	root	20	0	236432	5376	3640	S	0.7	0.1	0:31.63	ovs-vsmit+
1933	andy	20	0	368788	12080	7036	S	0.3	0.1	0:15.04	ibus-daem+
1958	andy	20	0	487852	31428	25344	S	0.3	0.4	0:05.93	ibus-ui-g+
1983	andy	20	0	533780	28692	22160	S	0.3	0.4	0:06.48	bamfdamon
2204	andy	20	0	542484	37360	29688	S	0.3	0.5	0:26.67	vmtoolsd
1	root	20	0	119888	6052	4052	S	0.0	0.1	0:05.45	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.06	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu+

● Using h1 ping h2

```
File Edit View Search Terminal Help
top - 09:28:48 up 6:42, 1 user, load average: 3.13, 2.29, 1.44
Tasks: 256 total, 5 running, 184 sleeping, 0 stopped, 0 zombie
%Cpu(s): 18.3 us, 27.1 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 54.6 si, 0.0 st
Mem: 8144668 total, 287408 free, 3201640 used, 4655620 buff/cache
Swap: 998396 total, 998128 free, 268 used. 4501280 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7	root	20	0	0	0	0	R	53.2	0.0	51:38.34	ksoftirqd+
16	root	20	0	0	0	0	R	40.9	0.0	51:47.51	ksoftirqd+
2403	andy	20	0	675632	45032	28372	R	30.2	0.6	1:26.77	gnome-ter+
26047	root	20	0	70552	19132	7540	R	24.3	0.2	0:11.96	mn
26495	root	20	0	6536	848	768	S	15.0	0.0	0:09.68	ping
940	root	20	0	507304	111172	56128	S	14.0	1.4	6:17.66	Xorg
23993	root	20	0	0	0	0	I	6.0	0.0	0:02.99	kworker/u+
25451	andy	20	0	5228736	735840	25724	S	6.0	9.0	1:34.12	java
26488	root	20	0	0	0	0	I	5.3	0.0	0:02.02	kworker/u+
2191	andy	20	0	1528328	117912	70380	S	4.7	1.4	5:52.33	compiz
8	root	20	0	0	0	0	I	0.3	0.0	0:18.77	rcu_sched
389	root	20	0	201540	10520	9204	S	0.3	0.1	0:23.19	vmtoolsd
1368	rtkit	21	1	183544	2972	2692	S	0.3	0.0	0:00.32	rtkit-dae+
1933	andy	20	0	368788	12080	7036	S	0.3	0.1	0:15.21	ibus-daem+
2434	andy	20	0	4690572	1.071g	19500	S	0.3	13.8	4:36.81	java
26475	andy	20	0	48988	3784	3100	R	0.3	0.0	0:00.71	top
1	root	20	0	119888	6052	4052	S	0.0	0.1	0:05.45	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.06	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu+

Step 6 : Switch 's packets

Flows for Device of:0000000000000001 (6 Total)

STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT	APP NAME
Added	0	2,255	40000	0	ETH_TYPE:arp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	447	2,255	40000	0	ETH_TYPE:bddp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	447	2,255	40000	0	ETH_TYPE:ldp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	4,491	1,565	50000	0	IN_PORT:1	imm([OUTPUT:2, OUTPUT:3], cleared:false)	*rest
Added	780,237,615	1,565	50000	0	IN_PORT:2	imm([OUTPUT:1, OUTPUT:3], cleared:false)	*rest
Added	791,433,509	1,565	50000	0	IN_PORT:3	imm([OUTPUT:1, OUTPUT:2], cleared:false)	*rest

Flows for Device of:0000000000000002 (6 Total)

STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT	APP NAME
Added	0	2,195	40000	0	ETH_TYPE:arp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	444	2,195	40000	0	ETH_TYPE:ldp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	444	2,195	40000	0	ETH_TYPE:bddp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	123,263	1,505	50000	0	IN_PORT:1	imm([OUTPUT:2, OUTPUT:3], cleared:false)	*rest
Added	757,741,230	1,505	50000	0	IN_PORT:3	imm([OUTPUT:1, OUTPUT:2], cleared:false)	*rest
Added	768,965,907	1,505	50000	0	IN_PORT:2	imm([OUTPUT:1, OUTPUT:3], cleared:false)	*rest

Flows for Device of:0000000000000003 (5 Total)

STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT	APP NAME
Added	0	2,235	40000	0	ETH_TYPE:arp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	447	2,235	40000	0	ETH_TYPE:ldp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	448	2,235	40000	0	ETH_TYPE:bddp	imm([OUTPUT:CONTROLLER], cleared:true)	*core
Added	773,019,009	1,543	50000	0	IN_PORT:2	imm([OUTPUT:1], cleared:false)	*rest
Added	784,263,946	1,544	50000	0	IN_PORT:1	imm([OUTPUT:2], cleared:false)	*rest

Step 7 : Explain why broadcast storm occurred

假設 H1 要傳送資料給 H2，

但 H1 的 ARP Table 裡沒有 H2 的 MAC Address，

因此 H1 送出一個 Broadcast ARP Request 想找 H2 的 MAC Address。

這個 Broadcast 送到 SW1，由於 SW1 在 Fa0/1 收到 Broadcast，

所以它會把 Broadcast 經由 Fa0/2 和 Fa0/3 送出去。

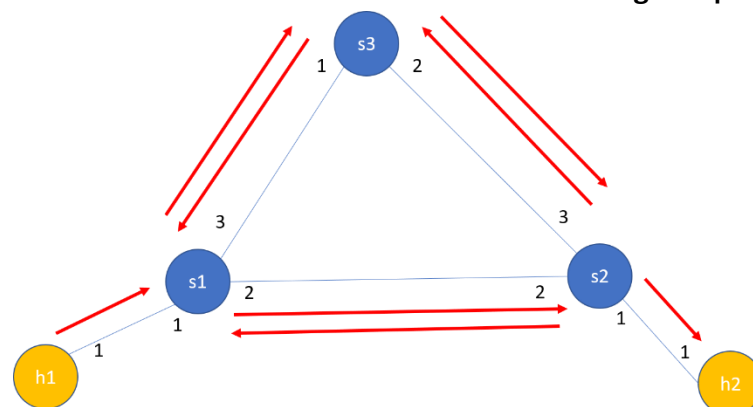
同樣地，SW3 在 Fa0/1 收到後會在 Fa0/2 送出去，

而 SW3 在 Fa0/2 收到後，就在 Fa0/1 和 Fa0/3 送出去，

於是 SW3 和 SW1 又收到來自 SW2 的 Broadcast，

然後它們也照樣在另一個出口送出，於是 Broadcast 不會停止地重覆輸送，

這就稱為 Broadcast Storm，會造成 Switching Loop 的問題。



Bonus

Control Plane: SDN Controller

Data Plane: Switch , Packet Forwarding

Step 1 :

Host 1 doesn't have Host 2's MAC

Then Host 1 send **ARP Request** to Switch

Step 2 :

Switch send **ARP Request** to Controller to find Host 2's MAC

Controller check its ARP table to find Host 2's MAC

Step 3 :

When Controller find Host 2's MAC then send **ARP Reply** to Switch

Step 4 :

Switch use **ARP Reply** to send Host 1 Host 2's MAC

Step 5 :

Host 1 send **ICMP Request** to Switch

Step 6 :

Switch check it's flow table then let **ICMP Request** go to Host 2

